УДК 004.382

В.С. Глухов, Р. Еліас
Національний університет "Львівська політехніка",
кафедра електронних обчислювальних машин

ЗАСОБИ ВІДЛАГОДЖЕННЯ ПРИСТРОЇВ ІЗ ВБУДОВАНИМ КОНТРОЛЕМ ДЛЯ ОБРОБЛЕННЯ ЕЛЕМЕНТІВ ПОЛІВ ГАЛУА $GF(2^M)$

На сучасному етапі математичною основою цифрових підписів є поля Галуа $GF(2^m)$. Розрядність елементів поля *m* може сягати понад 2000 бітів. Апаратна реалізація процесора для таких полів вимагає більш ніж мільйона транзисторів. Для збільшення надійності процесора він доповнюється вузлами вбудованого контролю. Перевірка роботи таких вузлів вимагає додаткових технологічних засобів. Особливості проектування цих засобів розглянуто у статті. Головною рисою таких засобів є можливість внесення помилок в VHDL-описи процесора з метою перевірки реакції на помилки вузлів вбудованого контролю.

Galois field is the mathematical basis for modern digital signatures. Field elements width may reach 2048 bits. Hardware realization of processor for such fields requires more than a million transistors. Processor is supplemented by concurrent error detection units to increase reliability. Testing of such units require additional technological tools. This article describes the design features of such tools. Tools main feature is ability to making mistakes in the VHDL-description of the processor.

Вступ. На сучасному етапі математичною основою цифрових підписів є поля Галуа $GF(2^m)$. Розрядність елементів поля *m* може сягати 2048 біт. Апаратна реалізація процесора для таких полів вимагає більш ніж мільйона транзисторів. Для збільшення надійності процесора він доповнюється вузлами вбудованого контролю. Перевірка роботи таких вузлів є важливою і актуальною задачею, яка вимагає додаткових технологічних засобів. Головною рисою таких засобів є можливість внесення помилок у VHDL-описи процесора.

Аналіз публікацій і окреслення проблеми. Сьогодні основним полем для проектувальників цифрових систем є ПЛІС. Традиційно описати проект на ПЛІС можна у вигляді:

– схеми;
– опису на мові опису апаратних засобів (*HDL*);
– графу автомата [1].

Найуніверсальнішим є опис *HDL*-мовами, сучасні засоби проектування забезпечують трансляцію описів у вигляді схем і графів мовою *HDL*. Також існують генератори *HDL*-описів стандартних вузлів цифрової техніки – генератори ядер [2].

Сучасні засоби проектування для створення опису цифрових пристроїв, придатного для проектування топології ПЛІС, використовують *HDL*-мови *VHDL* та *Verilog*. Для цілей моделювання додатково пропонуються засоби, реалізовані на основі мов високого рівня (*HLL*) типу *C/C++* (рис. 1): *SystemC, PLI/VPI/VHPI, SystemVerilog* [3], які разом із зручним інтерфейсом користувача забезпечують також атоматичне врахування результатів тестування на його послідовність (зворотний зв'язок на рис. 1).

Для деяких задач (обробка сигналів та зображень, прискорення вбудованих процсорів типу *Xilinx MicroBlaze* та *PowerPC*, цифрова обробка сигналів, шифрування та дишефрування згідно з 3*DES*, наукові обчислення, фінансові обчислення) існують засоби трансляції *C*-програм в *VHDL*-

описи [4–7], які розділяють процес проектування на 3 нитки: проектування апаратного забезпечення спеціалізованого процесора, проектування його інтерфейсу для зв'язку з керуючим процесором та розроблення програмного забезпечення керуючого процесора (рис. 2). При цьому для математичних обчислень під час розв'язання наукових задач використовуються стандартні бібліотеки math.h [8].

Для переходу від промодельованих мовоюі *C/C++* описів розробляють вузькоспеціалізовані транслятори [9]. Формується тенденція переходу від описів на рівні пересилання між регістрами (*RTL*, забезпечується засобами *HDL*) до описів на рівні електронних систем (*ESL*, забезпечується використанням мов високого рівня *HLL*) [10] з подальшою автоматичною генерацією *RTL*-описів і навіть автоматичною генерацією переліку зв'язків для проектування топології ПЛІС. Але сьогодні це лише тенденція.

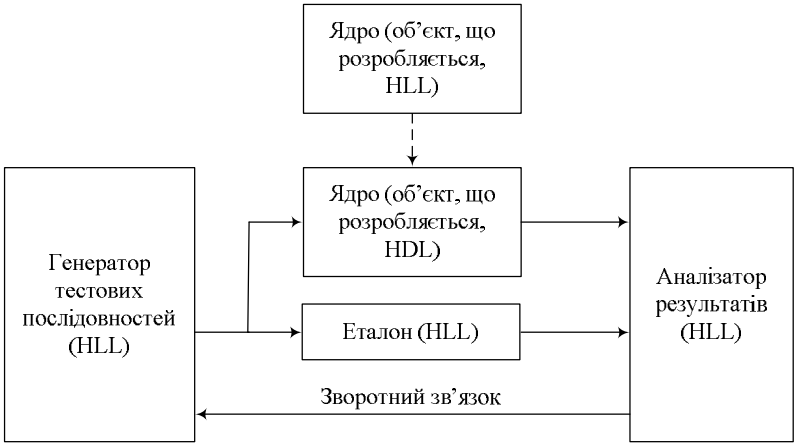

Рис. 1. Структурна схема процесу тестування та діагностики

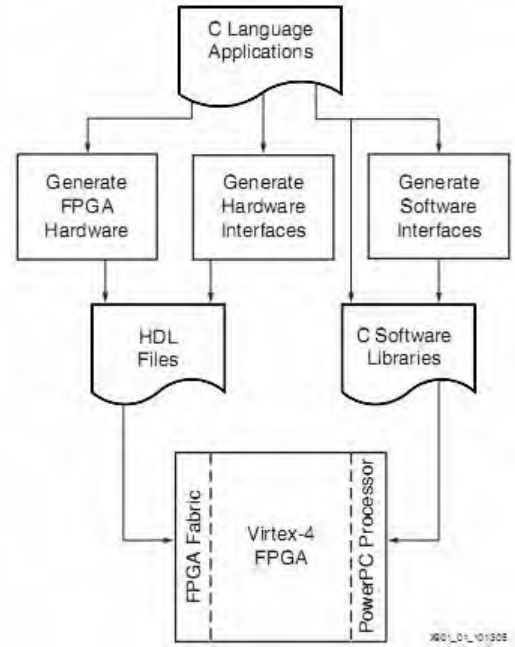

Рис. 2. Набір засобів CoDeveloper фірми Impulse Accelerated Technologies

Цифрові пристрої на ПЛІС складаються з протокольного універсального процесора та спеціалізованого процесора (рис. 3) [13].

Недоліком сучасних методів генерації описів функціональних вузлів є відсутність зв'язку між трьома етапами процесу проектування:

моделюванням алгоритму роботи вузла;

створенням опису функціонального вузла;

моделюванням роботи функціонального вузла.

Істотним недоліком також є відсутність засобів перевіряння вбудованих засобів контролю.

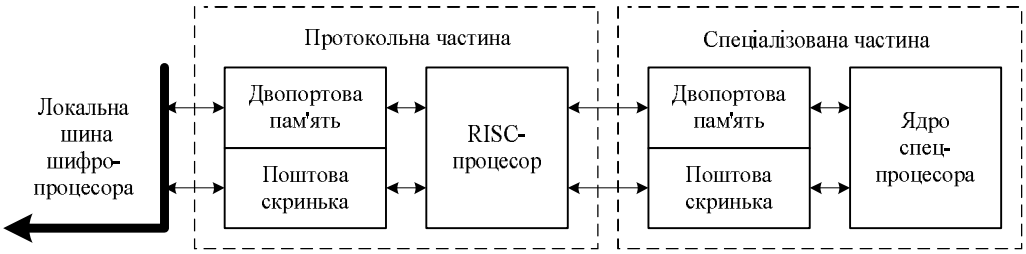

Рис. 3. Структура цифрового пристрою на ПЛІС

Мета роботи. Метою роботи є обґрунтування та проектування структури засобів відлагодження спеціалізованих процесорів, до складу яких входять вузли вбудованого контролю.

Метод генерації описів функціональних вузлів спеціалізованих процесорів. З метою покращення тестування описів функціональних цифрових пристроїв, які містять вузли вбудованого контролю, пропонується така послідовність проектування описів (рис. 4).

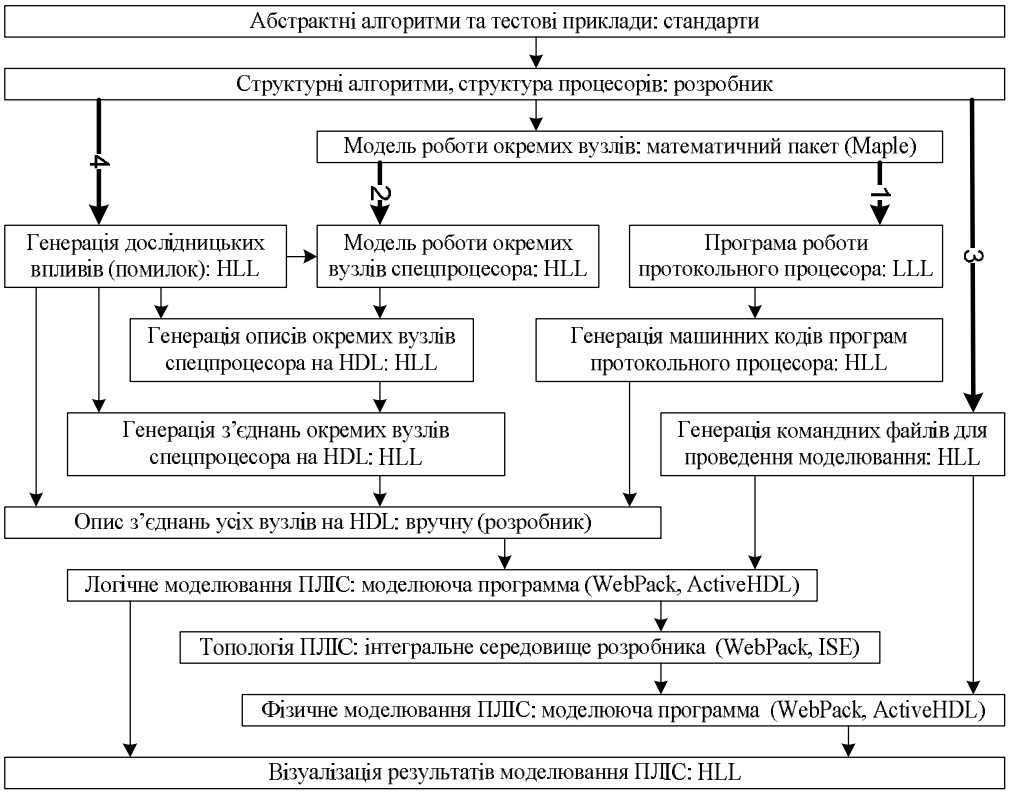

Рис. 4. Послідовність проектування описів функціональних вузлів

Особливість цього підходу полягає у тому, що:

процес проектування розбивається на чотири паралельні нитки: (1) програмування протокольного процесора, (2) проектування апаратного забезпечення спецпроцесора, (3) проектування технологічних засобів для моделювання та перевірки роботи спроектованих вузлів і (4) перевіряння засобів вбудованого контролю (цифри у дужках вказують на номер нитки на рис. 4);

проектування відбувається "згори-донизу-догори": від абстрактних алгоритмів до детальних описів окремих вузлів, а потім до перевіреної засобами моделювання топології кристала усього пристрою;

описи роботи вузлів мовою високого рівня (*HLL*-описів) створюються одночасно з розробленням програм-трансляторів цих описів мовою опису апаратного забезпечення (*HDL*-описів);

перевірка створених *HLL*-описів відбувається одночасно з перевіркою *HDL*-описів;

розроблювані *HLL*-описи та *HDL*-описи утворюють бібліотеку описів, елементи якої використовуються під час створення проекту загалом;

проектування може починатися з порожніми бібліотеками *HLL*- та *HDL*-описів;

з'єднання розроблених *HDL*-описів відбувається на етапі генерації *HDL*-опису вузла вищого рівня;

загальне під'єднання *HDL*-описів усіх вузлів в один проект відбувається в ручному режимі;

Цей метод передбачає володіння розробником мовою програмування високого рівня, низького рівня та мовою описів апаратних засобів.

Метод розрахований на проектування спеціалізованих вузлів. Результати його застосування (бібліотеки описів, програми-генератори) є спеціалізованими і не можуть бути використані для розв'язання задач іншого класу.

Вбудований контроль вузлів гарантоздатних комп'ютерних засобів. Для перевіряння вузлів вбудованого контролю система проектування повинна забезпечувати генерування спотворених описів проектованих вузлів [13].

Методи генерування спотворених описів ілюструє рис. 5 на прикладі перевіряння вузла вбудованого контролю складової частини ядра спецпроцесора – помножувача елементів поля Галуа $GF(2^m)$ у нормальному базисі.

У *VHDL*-опис пари (*i, j*) розрядів вхідної шини помножувача вставляється *VHDL*-опис генератора *G* помилок. Режим роботи генератора визначається сигналом керування *Mode*. Можливі такі режими роботи генератора помилок:

трансляція істинного значення вхідних сигналів I_0 та I_1 на виходи O_0 та O_1 (робота без імітації помилки);

генерування постійного 0 або 1 на одному або двох виходах O_0 та O_1 (імітація статичної помилки);

імітація закорочень двох вхідних сигналів I_0& I_1 або I_0vI_1 на двох виходах O_0 та O_1;

генерування правильного результату або 0 або 1 на одному або двох виходах O_0 та O_1. Значення 0 або 1 генерується з використанням псевдовипадкових кодів K_1 та K_2 (імітація динамічної помилки). Черговий біт коду K_1 визначає момент формування правильного результату або хибного, а черговий біт коду K_2 – значення хибного результату (0 або 1). Параметри псевдовипадкового коду (частка 0 або 1) можна змінювати.

При імітації статичних помилок та закорочень на кожнму тестовому вхідному значенні генератор помилок переміщається на один біт по вхідній шині при кожному наступному випробовуванні, доки випробовуваннями не буде пройдена уся шина.

При імітації динамічних випробовувань генератор залишається на одній позиції, доки не буде отримано бажаної кількості результатів. Після цього генератор також переміщається по шині на сусідню позицію.

Вбудований контроль множення у гауссівському нормальному базисі типу 2. Запропоновані засоби проектування було використано для досліджування вузла вбудованого контролю помножувача, який працює з елементами полів Галуа $GF(2^m)$, представленими у гауссівському нормальному базисі типу 2 [11, 14].

Рис. 6 містить схему помножувача та схему вузла вбудованого контролю, а також показує модель помилок у роботі помножувача, які досліджувалися: обрив на одному із входів помножувальної матриці *M*. Розглядалися дві ситуації: коли обірваний вхід сприймається як 0 (E=0) і коли обірваний вхід сприймається як 1 (E=1). Досліджувалася поведінка детектора помилок при формуванні цифрового підпису для тестового прикладу, наведеному у Додатку Б.2 [12]. Результати тестування наведено на рис. 7 (для випадку E=0) та рис. 8 (для випадку E=1).

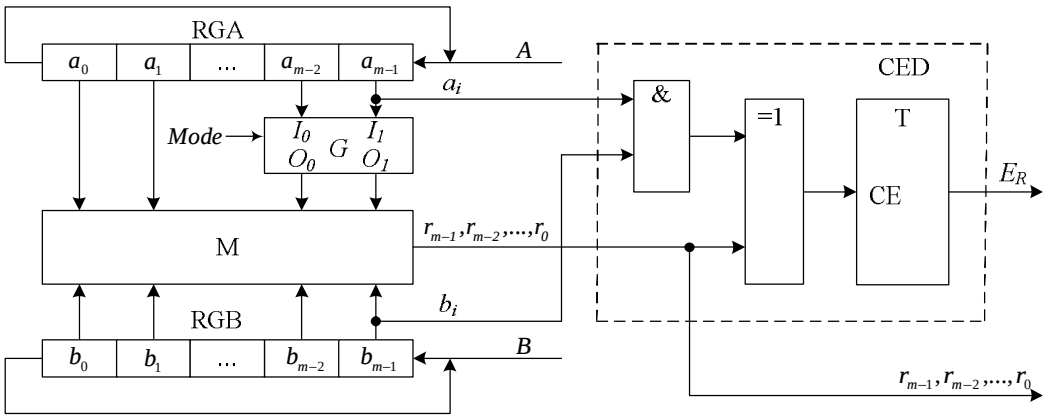

Рис. 5. Генерація спотворених описів

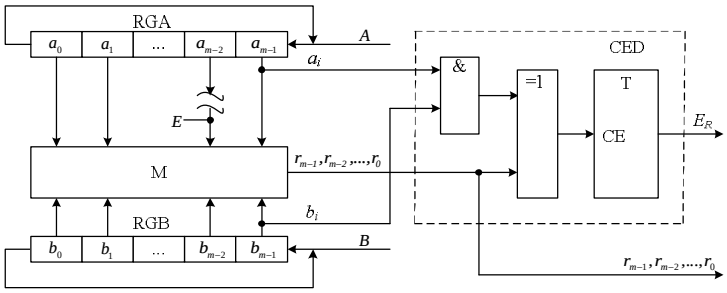

Рис. 6. Помножувач з CED (модель помилки №1)

Для цього ж тестового прикладу досліджувалася реакція детектора помилок при внесенні помилки до матриці M помножувача. Математична матриця M у розглянутому тестовому прикладі має розміри 173 рядка по 173 біти. Моделювалася поведінка детектора помилок при інверсії біта у 0-му рядку та у 172-му рядку матриці. Результати дослідження наведено на рис. 9 та рис. 10 відповідно.

Як видно з наведених графіків, кількість послідовних операцій множення при обробленні одного цифрового підпису сягає 8983. З них більше ніж у 2000 детектор виявляє помилки.

Імовірність виявлення помилки помножувача при обробленні цифрових підписів. Позначимо як p – імовірність виявлення помилки у роботі помножувача. Імовірність невиявлення помилки $q = 1 - p$. Для k послідовних множень імовірність невиявлення помилки $Q = q^k = (1\text{-}p)^k$. Імовірність виявлення помилки у послідовності множень

$$P = 1 - Q = 1 - (1 - p)^k.$$

Для $p = 0{,}5$, $k = 8983$ маємо $Q = 0{,}5^{8983} = 1/2^{8983}$ (імовірність підбору 173-бітного особистого ключа в наведеному прикладі становить $I = 1/2^{173}$, тобто, набагато більша ніж імовірність невиявлення помилки помножувача при обробленні цифрових підписів).

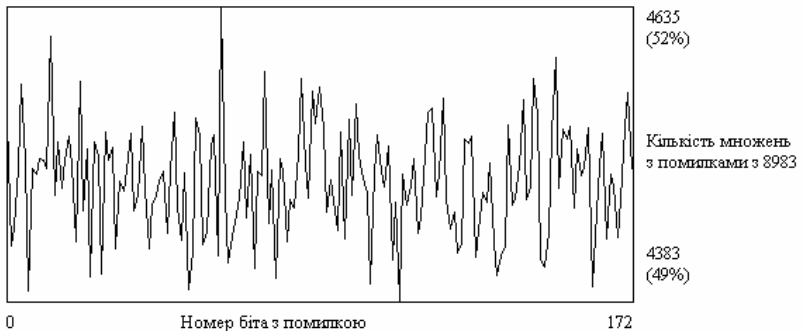

Рис. 7. Модель помилки № 1 (E = 0)

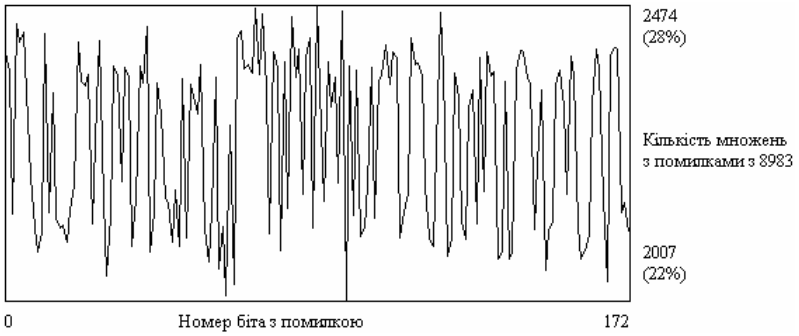

Рис. 8. Модель помилки № 1 (E = 1)

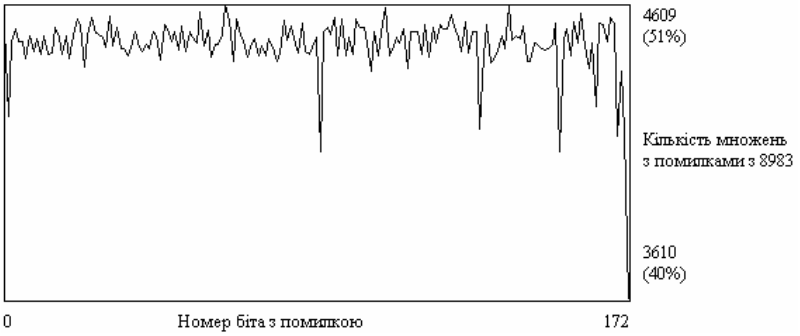

Рис. 9. Модель помилки № 2 (помилка в 0-му рядку матриці M)

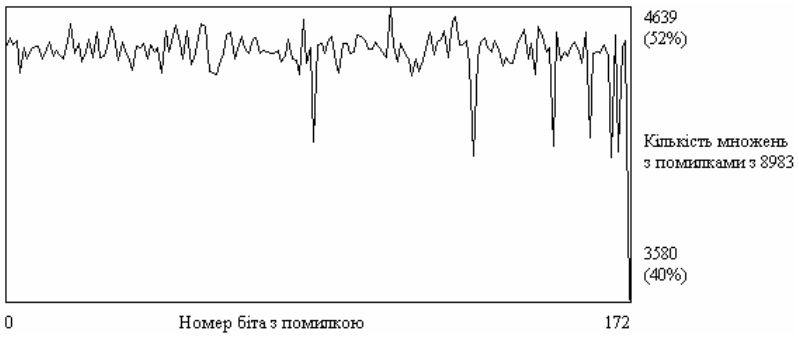

Рис. 10. Модель помилки № 3 (помилка в 172-му рядку матриці M)

Висновки. У роботі запропоновано і обгрунтовано структуру засобів відлагодження спеціалізованих процесорів, до складу яких входять вузли вбудованого контролю. На відміну від відомих структур запропонована додатково забезпечує проектування технологічних засобів для моделювання та перевірки роботи спроектованих вузлів і перевіряння засобів вбудованого контролю. Для цього до складу засобів введений генератор помилок, який забезпечує перевіряння реакції засобів вбудованого контролю на статичні та динамічні помилки. У статті визначено характеристики такого генератора. Також у статті обґрунтовано доцільність використання у пристроях оброблення цифрових підписів помножувача елементів поля $GF(2^m)$ з вбудованим контролем, який виявляє усього 50% помилок множення. При цьому імовірність виявлення помилки при обробленні цифрового підпису дуже близька до 100%. Цей метод вбудованого тестування рекомендується використовувати при роботі у нормальному базисі типу 2 полів $GF(2^n)$, які рекомендовані до використання стандартом ДСТУ 4145-2002.

1. Грушвицкий Р.И., Мураев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.: ил. 2. Clive Maxfield. The Design Warrior's Guide to FPGAs: Devices, Tools and Flows. Elsevier, 2004. ISBN 0750676043, 9780750676045. 542 p. 3. Ю. Потапов. Продукты компании Aldec для проектирования цифровых устройств. "Электронные компоненты" №1' 2005. 4. Glenn Steiner, Kunal Shenoy, Dan Isaacs, David Pellerin. How to accelerate algorithms by automatically generating FPGA coprocessors. August 09,

2006. http://www.drdobbs.com/embedded-systems/191901647 5. David Pellerin, Kunal Shenoy. C-Language techniques for FPGA acceleration of embedded software. Embedded Systems Conference Silicon Valley 2006. http://www.eetimes.com/design/programmable-logic/4014817/C-Language-techniques-for-FPGA-acceleration-of-embedded-software 6. David Pellerin, Kunal Shenoy. C-Language techniques for FPGA acceleration of embedded software. Embedded Systems Conference Silicon Valley 2006. http://www.eetimes.com/design/programmable-logic/4014817/C-Language-techniques-for-FPGA-acceleration-of-embedded-software 7. Kunal Shenoy. Accelerating Software Applications Using the APU Controller and C-to-HDL Tools. XAPP901 (v1.0) December 16, 2005. 8. Michael Kreeger, Brian Durwood. Accelerating floating-point designs on FPGAs using math.h function. MILITARY EMBEDDED SYSTEMS. July/august 2010. pp. 36-39. ISSN: Print 1557-3222. © 2010 OpenSystems Media. © 2010 Military Embedded Systems 9. Ковалев А.В. Разработка метода построения VHDL-описаний СФ-блоков для повторного использования в системах обработки изображений на основе описаний на языке SystemC // Актуальные проблемы твердотельной электроникии микроэлектроники: Труды девятой международной научно-технической конференции. – Таганрог, 2004. 10. Ron Wilson. Electronic-system-level design: is there fire beneath the smoke? EDN Europe magazine. October 2008, pp.25-31. 11. Глухов В.С. Вбудований контроль множення в гауссівському нормальному базисі типу 2 полів Галуа GF(2^m) // Науково-технічний журнал "Радіоелектронні і комп'ютерні системи 6(47). Національний аерокосмічний університет ім. М.Є. Жуковського "Харківський авіаційний інститут". – Харків: ХАІ. – 2010. – С. 255 – 259. 12. ДСТУ 4145-2002. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння. Київ. Державний комітет України з питань технічного регулювання та споживчої політики. 2003. 13. Глухов В.С., Еліас Р. Вбудований контроль спецпроцесорів для оброблення цифрових підписів // Вісник Нац. ун-ту "Львівська політехніка" "Комп'ютерні науки та інформаційні технології". – 2010. 14. Глухов В.С., Еліас Р. Ефективність вбудованого контролю пристоїв обробки електронних цифрових підписів // Международный научно-технический семинар "Современные проблемы прикладной математики, информатики и автоматизации". г. Севастополь. Севастопольский национальный технический университет, 04–07 октября, 2010 г.